WHAT SHOULD STUDENTS GET FROM CALCULUS? (AND HOW CAN WE PROVIDE IT?)

FRANK QUINN

In his September *Doceamus* column [1], Keith Stroyan takes on this question and reports success with extended exploration of applications. Experiences with a different population of students have led me in rather different directions on practical levels, but with important commonalities.

I back up a bit for perspective. The actual title of Stroyan's column is *Why do so many students take calculus?*. But the honest answer to this is "because it is required in the curriculum", and the real problem is that traditional calculus courses do not serve students particularly well. Stroyan actually addresses a variation: "We've got them here; what is the best use we can make of this opportunity?" I think his answer is a good one for many circumstances. A more pointed variation is "Calculus is emphasized at least partly for historical reasons; would a different topic work better?" Stroyan's explicit comment is that calculus is good because it gives quick access to rich and varied applications. His approach addresses this concern in other ways: it has less emphasis on lectures and the nitty-gritty of calculus, and exploring applications often brings in methods from other areas.

Nearly all of my students are in science and engineering, and this has led me to a more mission-oriented version of the question. Namely, "What do students need, and what are the most valuable things they get?" My main concern is with course design. Scientists and engineers do still need a foundation in calculus, but I see calculus as a setting rather than a goal, and even for this group I don't think "knowledge of calculus" is among the most valuable outcomes. The next three sections describe other important goals.

Complex rules and accuracy. It is a vital skill in science and engineering to be able to work accurately with complex rule-based systems. I feel that it is part of our job to develop this: calculus is certainly the best training ground in the current curriculum because the rules are realistically complicated, but are clear and concise and feedback is quick and accurate. This skill is also transferrable to many more domains than any specific content. But this is a skill that my students certainly don't have when they get here.

Most high-school programs have de-emphasized rule skills in favor of "understanding" and working intuitively. If you can "see" the problem it should be easy. Calculator use has replaced a lot of rule-based work and attendant skills. AP calculus is a partial exception, but it is test-driven with greatly simplified rules used mechanically on short, routine problems. Low skill levels give mediocre results even on simplified problems, but this has been compensated for by generous partial and extra credit, and curved grades.

Date: Draft Oct 20 2011.

Given all this, I feel that helping my students develop disciplined rule skills is the most important service I can provide. An implication for course design—again, for students in science and engineering—is that corners should not be cut. Do all the standard techniques of integration, with the full set of elementary functions, to provide enough complexity to require careful and systematic use. Finally I think this is the time to get real about getting things right. High-tech employers don't give 'A's for work that is 90% right. I expect students to get up to speed, rather than reducing the speed to their comfort zone. Their poor preparation makes this is a serious challenge, but an important one and most of them rise to it.

Abstract and symbolic work. Technical challenges in science and engineering are getting more difficult, while dealing with numbers is getting easier. A consequence is that work on an abstract and symbolic level—even if only to organize numerical work—is increasingly important. But these skills are also declining. Some of my students have trouble with any problem whose answer is not a number: they can handle circles of radius 3, but simple problems with circles of radius 'r' are foreign territory.

Again, I feel my students are better served if I can help develop these skills. My examples and problems usually have symbolic parameters, and I emphasize what these reveal about scaling, optimization, and error analysis. I usually use exact arithmetic. This preserves structure (π and $\sqrt{2}$ don't disappear into decimals), and is half-way to symbolic work. Again this is a challenge, and quite a few students need remediation before it is accessible, but it is reasonable to expect them to get it before attempting a science and engineering curriculum.

Applications. Applications provide opportunities for students to exercise their skills and see the methods in action. However applications do not have to be to physical problems, and in fact I find most physical applications unsatisfactory.

- It is a good idea to plug in numbers from time to time but it destroys a lot of functionality. Printing out web pages can also be a good thing, but it kills the functionality of links. In particular, extended numerical applications do not exercise the most important skills.
- Most of these students have specific interests. Applications that address their interests will duplicate material done in more depth in other courses. Applications that don't address their interests don't engage them.
- Superficial applications are usually little more than vocabulary (replace 'velocity' with the first derivative, etc.). These are worth mentioning, but as testable material are not a good use of their time.

On the other hand working a bit more abstractly and symbolically opens up mathematical topics to explore. These are, in effect, applications that are both mission-related and quickly accessible because techniques and terminology are already in place. I also find that laying groundwork for mathematical applications helps organize and sharpen presentations.

Resource constraints. Unfortunately, one more version should be addressed: "Even if we do figure out what students need, can we afford to provide it?" I individually, and the department at Virginia Tech collectively, have tried many things that improved learning but that had to be abandoned because they required unsustainable levels of faculty overtime. These include group projects along the lines described in [1]. My suggestions here are also problematic. They don't directly cost more, but increasing expectations increases failure rates unless individual help is provided, and appropriate help would definitely be over-budget. I have also omitted quite a few effective strategies because they are impossible without individual help.

We should remember that per-student resource levels were established at a time when we only lectured and gave tests. In many places they have declined substantially below this, and huge classes taught by adjuncts are increasingly common. In this climate any innovation that costs more is a dead end. Real impact in first- and second-year courses will require innovations whose resource requirements are competitive with huge classes taught by adjuncts. So far an educational approach has been taken: "discipline stifles creativity, so let a thousand flowers bloom". Unsurprisingly, we have gotten education-quality outcomes: the thousand flowers bloomed and wilted, and very few students are better off. Maybe it is time to get real about getting it right, perhaps with a science and engineering approach: "no discipline, no results".

The sticking point is that, as far as I can see, the only way to both innovate *and* reduce costs is to give up traditional classrooms. This can work: we now have more than 10,000 students per semester taking lower-level computer-based courses. Moreover the unit costs are enough below the huge-class-with-adjuncts cutoff that Stroyan's group projects, or the individual help I feel is so important, would not put it over-budget.

There is still a big challenge: developing high-quality courseware and tests that would provide an environment for other innovations. Most courseware now follows the classroom model, which is a bit like teaching calculus without regard to student needs. We need materials much better adapted to individual use. Real success will also require sophisticated adjustments in the content. My experience is that this is a job for mathematicians, not educators[1].

Summary. Stroyan suggests a kinder, gentler calculus with extended projects on physical applications. I propose a more rigorous course with fewer physical applications. How can I see these as basically similar?

Both of us are concerned that traditional calculus courses do not serve students particularly well. We both feel—for rather different reasons—that calculus is a good setting, and the real problem is the traditional format. In particular, calculus is not the main learning goal even in a calculus course. We both believe that better goals depend on student needs; the differences in our specific approaches simply reflect concern with different student populations. And we have both concluded that serving students well will require activity—again different in detail—outside traditional classroom settings.

A final similarity is that wide implementation of either approach is serously limited by resource constraints. They might best be seen as examples of enrichments and student-specific variations that would be possible with high-quality computer-based courses.

References

[1] Stroyan, Keith; *Why do so many students take calculus?*, AMS *Notices* **58** (September 2011) pp. 1122–1123.

[1]See the essays at http://www.math.vt.edu/people/quinn/education/ for extensive discussion of these issues.